CUDA based Level Set Method for 3D Reconstruction of Fishes from Large Acoustic Data

Sharma, Ojaswa; Anton, François

Published in:
International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

CUDA based Level Set Method for 3D Reconstruction of Fishes from Large Acoustic Data

Ojaswa Sharma
Department of Informatics and Mathematical Modelling
Technical University of Denmark, Denmark
os@imm.dtu.dk

François Anton
Department of Informatics and Mathematical Modelling
Technical University of Denmark, Denmark
fa@imm.dtu.dk

ABSTRACT

Acoustic images present views of underwater dynamics, even in high depths. With multi-beam echo sounders (SONARs), it is possible to capture series of 2D high resolution acoustic images. 3D reconstruction of the water column and subsequent estimation of fish abundance and fish species identification is highly desirable for planning sustainable fisheries. Main hurdles in analysing acoustic images are the presence of speckle noise and the vast amount of acoustic data. This paper presents a level set formulation for simultaneous fish reconstruction and noise suppression from raw acoustic images. Despite the presence of speckle noise blobs, actual fish intensity values can be distinguished by extremely high values, varying exponentially from the background. Edge detection generally gives excessive false edges that are not reliable. Our approach to reconstruction is based on level set evolution using Mumford-Shah segmentation functional that does not depend on edges in an image. We use the implicit function in conjunction with the image to robustly estimate a threshold for suppressing noise in the image by solving a second differential equation. We provide details of our estimation of suppressing threshold and show its convergence as the evolution proceeds. We also present a GPU based streaming computation of the method using NVIDIA's CUDA framework to handle large volume data-sets. Our implementation is optimised for memory usage to handle large volumes.

Keywords: 3D reconstruction, Level Set method, Acoustic images, Noise suppression, GPU, CUDA.

1 INTRODUCTION

One of the areas of interest in fisheries research is to reconstruct moving schools of fishes in a water column. Presence of strong speckle noise is a major problem in segmenting acoustic images. This makes selection of a threshold for binary segmentation very difficult [23]. The main contribution of this paper is to design a level set formulation that is well suited to reconstruct fishes from acoustic images captured using multi-beam echo sounders. The evolution of the level set equation is coupled with a solution of another differential equation that effectively removes the noise, enabling the level set to converge to the objects of interest in the image.

Although hyperspectral underwater imagers provide better imagery of underwater scenes, multibeam echo sounders are not outdated by them and complement them very well. The hyperspectral optical devices have very short range (sometimes less than 1 m in the North sea and in the Baltic). Acoustic sensors are still widely used in underwater surveys.

Speckle noise in acoustic images is generally modelled by the Rayleigh distribution [6, 5]. Quidu et al. [23] estimate an optimal image filter size to compute an estimate of a good threshold by pixel correlation. Gagnon [7] shows numerical results of a wavelet domain based method for noise removal. Chen and Raheja [4, 1] show a wavelet lifting based method where the spatial correlation of acoustic speckle noise is broken by multiresolution analysis. In another approach to use wavelet based methods, Isar et al. [10] present a Bayesian-based algorithm. In a novel attempt to use the Markov Random Field (MRF) to segment acoustic images, Mignotte et al. [17] use an unsupervised scheme by employing an iterative method of estimation called Iterative Conditional Estimation (ICE). The authors used a maximum likelihood estimation to compute the MRF prior model.

Krissian et al. [11] provide a variation of the anisotropic diffusion process [22] constrained by speckle noise model. Anisotropic diffusion provides an intelligent way to perform diffusion without affecting prominent edges in an image.

Level set based methods have been shown to successfully restore noisy images [24]. Osher and Rudin [20] developed shock filters for image enhancements. Malladi and Sethian [16] have shown image smoothing and enhancement based on curvature flow interpretation of the geometric heat equation. In a more recent approach to use level set methods for acoustic image

segmentation, Lianantonakis and Petillot [14] provide an acoustic image segmentation framework using the region based active contour model of Chan and Vese [2]. The authors comment that the level set based model has good regularisation properties similar to those of a Markov random field [14].

A relevant work by Balabanian et al. [1] shows an interactive tool for visualization of acoustic volumetric data using a well known volume visualization technique called Ray-casting. Authors in [1] develop a tool for manual selection by region growing and visualization of moving fish schools using graphics hardware. The work presented in this paper is intended to develop mathematical models to automatically extract meaningful features from acoustic data with no user interaction. This work does not provide any tool for visual analysis, rather it presents a computational framework for noise suppression and 3D reconstruction.

This paper concentrates on using the level set methods for simultaneous suppression of noise and 3D reconstruction of relevant features. We limit features of interest to fishes from acoustic images and provide a level set based framework for acoustic image segmentation. Image restoration techniques based on level set evolution are generally oriented to segment the image or to remove noise from it. Work by Lianantonakis and Petillot [14] is closest to our approach since they use active contours using Mumford-shah functional for seabed classification, but together with extraction of Haralick feature set for textural analysis. Our method differs from theirs since it is not possible to rely on texture based classification in the absence of any specific textures in the image.

Since acoustic data resulting from marine surveys can result in gigabytes of information, we employ GPU (Graphics Processing Unit) based computations for 3D reconstruction. The GPU is not very suitable for data intensive applications due to unavailability of large memory on commodity hardware. A number of publications suggest schemes to circumvent this situation by performing computations in a streaming manner [13, 12, 9], but most of the implementations process 2D sections to generate a 3D reconstruction. We present a Level Set method implementation with computations performed entirely in 3D using the 3D textures (read only) available to the CUDA 2.0 framework. CUDA (Compute Unified Device Architecture) is a parallel programming model and software environment designed to develop application software that transparently scales its parallelism to leverage the increasing number of processor cores on the GPU. It allows programming computationally intensive algorithms to take advantage of the available graphics hardware. Our method is streaming and is optimised for memory usage, consuming only twice the CPU memory of the input volume.

The paper is organised as follows. In Section 2, we present the preliminaries on the active contour model. In Section 3, we present the work of Chan and Vese [2] on minimising the Mumford-Shah functional in images. In Section 4, we present our work on the noise suppression model which is solved together with the level set equation. In Section 5, we present our CUDA implementation for 3D reconstruction of the fishes based on the parallelisation of the results of Section 4. In Section 6, we present the experimental results. Finally we conclude the paper in Section 7.

2 BACKGROUND

Let an image $I(x,y)$ be defined on a bounded open subset $\Omega : \{(x,y)|0 \leq x,y \leq 1\}$ of $\mathbb{R}^2$, with $\partial\Omega$ as its boundary. I takes discrete values between 0 and $(2^n - 1)$ where n is the number of bits used to store intensity. The basic idea in active contour model is to evolve a curve $C(s) : [0,1] \rightarrow \mathbb{R}^2$ by minimising the following energy functional [19]:

$$E(C) = \alpha \int_0^1 |C'|^2 \, ds + \beta \int_0^1 |C''| \, ds - \lambda \int_0^1 |\nabla I(C)|^2 \, ds,$$

where, α, β, and λ are positive parameters. In the above energy functional, the evolution of curve C is controlled by the internal energy (first two terms that define the smoothness of the curve) and the external energy (the last term that depends on the edges present in the image). The curve C can be represented by an implicit function ϕ, $C = \{(x,y)|\phi(x,y) = 0\}$, where the evolution of C is given by the zero level curve at any time t of the function $\phi(x,y,t)$.

With this formulation, an edge detector is defined as a positive decreasing function $g(\nabla I)$ based on the gradient of image [22] such that

$$\lim_{|\nabla I| \rightarrow \infty} g(\nabla I) = 0$$

Therefore, the zero level curve evolves in the normal direction and stops at the desired boundary where g vanishes.

Evolving the curve C in normal direction amounts to solving the partial differential equation (PDE) [21]

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| F \tag{1}$$

with the initial condition $\phi(x,y,0) = \phi_0(x,y)$, where $\phi_0(x,y)$ is the initial contour. Motion by mean curvature allows for cusps, curvature and automatic topological changes [21, 3]. This results in the speed function $F = div\left(\frac{\nabla \phi}{\|\nabla \phi\|}\right)$ in terms of the curvature of ϕ

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| div\left(\frac{\nabla \phi}{|\nabla \phi|}\right), \phi(x,y,0) = \phi_0(x,y)$$

where $div(\cdot)$ is the divergence operator.

3 MINIMISING THE MUMFORD-SHAH FUNCTIONAL IN IMAGE

Chan and Vese [3] provide an alternative approach to the edge based stopping criterion. The authors suggest the stopping term based on Mumford-Shah segmentation techniques [18]. The motivation behind using this alternative stopping term is that in many cases, the edges in an image are not very well defined. Either it is ambiguous to position the edges across the gradient due to smoothly varying intensities [3] or it is difficult to select prominent edges due to presence of noise (as in the case of acoustic images). The method of Chan and Vese [3] is minimisation of an energy based segmentation. Assuming that the image I is composed of two regions of piecewise constant intensities of distinct values I^i and I^o, and that the object of interest is represented by I^i, we define the curve C to be its boundary. Using the Heaviside function H, and the Dirac-Delta function δ_0,

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}, \delta_0(z) = \frac{d}{dz} H(z)$$

the energy functional is formulated as

$$\begin{aligned} E(c_1, c_2, C, t) = & \mu \int_\Omega \delta_0(\phi(x,y,t)) |\nabla \phi(x,y,t)| \, dx\, dy \\ & + \nu \int_\Omega H(\phi(x,y,t)) \, dx\, dy \\ & + \lambda_1 \int_\Omega |I(x,y) - c_1|^2 \, dx\, dy \\ & + \lambda_2 \int_\Omega |I(x,y) - c_2|^2 \, dx\, dy \end{aligned} \quad (2)$$

where, $\mu \geq 0$, $\nu \geq 0$, $\lambda_1, \lambda_2 > 0$ are fixed parameters. c_1 and c_2 are average intensity values inside and outside C. The constants c_1 and c_2 can also be written in terms of I and ϕ

$$c_1 = \frac{\int_\Omega I(x,y) H(\phi(x,y,t)) \, dx\, dy}{\int_\Omega H(\phi(x,y,t)) \, dx\, dy}, \quad (3)$$

$$c_2 = \frac{\int_\Omega I(x,y) (1 - H(\phi(x,y,t))) \, dx\, dy}{\int_\Omega (1 - H(\phi(x,y,t))) \, dx\, dy} \quad (4)$$

The variational level set approach gives the following Euler-Lagrange equation [3]

$$\frac{\partial \phi}{\partial t} = \delta_\varepsilon(\phi) \left[\mu \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} - \nu - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2 \right] \quad (5)$$

with the initial condition, $\phi(x,y,0) = \phi_0(x,y)$ and

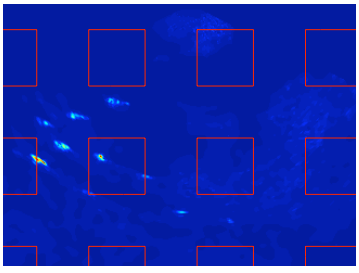
$$\delta_\varepsilon(z) = \frac{\partial}{\partial z} H_\varepsilon(z) = \pi^{-1} \varepsilon^{-1} \left(1 + \frac{z^2}{\varepsilon^2} \right)^{-1} \quad (6)$$

where, the regularised one-dimensional Heaviside function is given by:

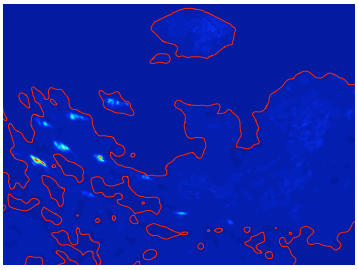
$$H_\varepsilon(z) = \frac{1}{2} \left(1 + \frac{2}{\pi} \tan^{-1} \left(\frac{z}{\varepsilon} \right) \right).$$

Despite the fact that this model has advantages over the edge based model in that it is able to detect boundaries with smoothly varying intensities and blurred edges, the main limitation comes from the fact that it can only discriminate regions with different mean intensities [14]. In particular, strong textures pose a problem with this approach. Lianantonakis and Petillot [14] solve this problem by extracting the Harlick feature set based on the co-occurance matrix.

The acoustic images considered by Lianantonakis and Petillot [14] are of the seabed. Such images show strong textural variations of the bottom surface of the sea. In this paper, we restrict ourselves to acoustic images of freely swimming fishes. While such images are also corrupted by speckle noise, they do not show specific textural patterns. Figure 1(a) shows part of such an image where the fish cross sections are discriminated by very high intensities compared to the background. The presence of reflectance from air bubbles mixing into water, also contribute to the noise. While working with level sets, a standard procedure is to keep ϕ to a signed distance function [19]. A direct application of the level set equation given by equation (5), with $\phi(x,y,0) = 0$ initialised to set of squares regularly distributed over the image, shows that the evolution of the level set eventually stops at the wrong place (see figure 1(b)). Furthermore, lack of any specific textural patterns leads us to formulate a successive noise suppression scheme where the Mumford-shah energy functional is minimised while simultaneously removing noise from the image. The later aids in fast convergence of the level curve in our formulation.



(a) Initialisation contour.



(b) Result at convergence.

Figure 1: Application of the level set equation (5) .

4 NOISE SUPPRESSION MODEL

As discussed before, acoustic images suffer from heavy speckle noise. At first thought, it might sound reasonable to apply a global threshold to the image to get rid of the noise. However, this is not a plausible option since for a particular chosen threshold there might be echo intensities of fishes lower than it and therefore such a threshold will result in loss of information [25, sec. 5.4.6, 6.3]. An adaptive threshold might also not provide a solution since the speckle has a high local intensity, and therefore could show false positives. Therefore, we resort to global energy minimising methods to suppress noise.

Considering the image I to be time varying, the basic idea behind noise suppression is to solve the following equation as an update step to the level set equation resolution in a single pass:

$$\frac{\partial I(x,y,t)}{\partial t} = k \cdot \max(0, \hat{c} - I(x,y,t)) \tag{7}$$

where k is a constant and $\hat{c}$ is a scalar parameter that is computed as an optimal threshold at any time step t based on $\phi(x,y,t)$.

The computation of $\hat{c}$ is based on the bounded subset I^i given by

$$I^i(x,y,t) = I(x,y,t) \cdot H_\varepsilon(\phi(x,y,t)).$$

The values given by the set I^i are used to compute the weighted median [26] as shown in algorithm 1 which is used as $\hat{c}$ at that particular time step t.

Input: $I(x,y,t)$, $H_\varepsilon(x,y,t)$
Output: $\hat{c}$
$V = \{v_i : v_i = I(x,y,t),\ x \in [1,l],\ y \in [1,m],\ i \in [1,n],\ n = l \cdot m\}$
$W = \{w_i : w_i = H_\varepsilon(x,y,t),\ x \in [1,l],\ y \in [1,m],\ i \in [1,n],\ n = l \cdot m\}$
Sort V in ascending order
$W \leftarrow W \backslash \{w_z\}\ \forall w_z = 0$
$V \leftarrow V \backslash \{v_z\}, \{v_z : v_z \in V, \forall z \text{ where } w_z = 0\}$
$S \leftarrow \sum_{k=1}^{n} w_k,\ w_k \in W$
Find largest index i such that $\sum_{k=1}^{i} w_k \leq \frac{S}{2}, w_k \in W$
Find smallest index j such that $\sum_{k=j}^{n} w_k \leq \frac{S}{2}, w_k \in W$
Median $M = \{v_i, v_j\}$
$\hat{c} \leftarrow \min(v_i, v_j)$
Algorithm 1: Computation of weighted median

The use of median filtering to remove noise is not new in image processing [8, 15]. We now show that the estimate of $\hat{c}$ based on the weighted median is a good approximation for the grey-level threshold that separates the noise from the signal, and is robust in a way that the evolution of the level set converges with increasing t.

$H_\varepsilon(z)$ attains values close to zero for regions outside C and values close to one inside C. In fact, $\lim_{z\to\infty} H_\varepsilon(z) = 1.0$ and $\lim_{z\to-\infty} H_\varepsilon(z) = 0.0$. At the start of level set evolution, I^i covers most of Ω and therefore, $H_\varepsilon(z)$ attains values close to one for most of the intensity values. This results in computation of $\hat{c}$ which is equivalent to an unweighted median for values in I^i. A median is the central point which minimises the average of absolute deviations. Therefore, a median better represents the noise level when the data contains high intensity values that are fewer in number, and a majority of intensity values that correspond to the noise. As a result, the initial iterations of the solution suppress the intensity values that are less than the median to a constant level (the median itself). One should expect the median value to increase as the level set contracts, but since we use a regularised Heaviside function as weight for the intensity values, the weighted median converges to zero since most of I contains intensity values of zero with near-zero weight.

Other variations of estimation of $\hat{c}$ are certainly possible, but we find that a weighted median based approach results in effective noise removal with very small information loss. For instance, a value of $\hat{c}$ taken to be c_1, the mean intensity inside C, does a similar suppression but with a high signal loss compared to the former. Furthermore, the mean does not converge as fast as the median does and might result in relatively higher values for large fish cross sections. It must be noted however, that the computation of the median is costly as compared to that of the mean.

5 CUDA IMPLEMENTATION FOR 3D RECONSTRUCTION

Equation (5) can be solved by discretization and linearization in ϕ[3]. Discretization of equation (7) in I gives

$$\begin{aligned}\frac{I_{n+1}(x,y) - I_n(x,y)}{\Delta t} &= k \cdot \max(0, \hat{c} - I_n(x,y)) \\ &= \begin{cases} 0, \text{if } I_n(x,y) \geq \hat{c} \\ k \cdot (\hat{c} - I_n(x,y)), \text{otherwise} \end{cases}\end{aligned} \tag{8}$$

With $k = \frac{1}{\Delta t}$, and $t_{n+1} = t_n + \Delta t$. The above time discretization yields the following

$$I_{n+1}(x,y) = \begin{cases} I_n(x,y), & \text{if } I_n(x,y) \geq \hat{c} \\ \hat{c}, & \text{if } I_n(x,y) < \hat{c} \end{cases} \tag{9}$$

Acoustic images captured by echo-sounders are generally taken as planar image scans by moving the echo-sounder in one direction, thereby sweeping a volume.

Let us denote individual images as $I(x,y,\tau)$ for images taken after every $\delta\tau$ time interval. A volume is constructed by stacking these individual images in sequence and applying geometric correction for distance $\delta\tau(v)$ between individual slices, where v is the instantaneous speed of the instrument (the current data was captured with constant unidirectional instrument velocity). It must also be noted that the individual acoustic images are obtained from a set of acoustic intensity signals along beams by a polar transformation. The level set equations for curve evolution in $\mathbb{R}^2$ extend uniformly to surface evolution in $\mathbb{R}^3$. The second differential equation also holds true for noise suppression in a volume. Therefore, it is possible to reconstruct 3D moving fishes with the level set evolution of these equations combined.

Processing a huge dataset demands that a minimum of memory is consumed. We propose to keep two volumes in the host memory, one for the intensity values (I) and the other for the signed distance function (the implicit function, ϕ). The CPU manages the memory scheduling by dividing the volumes into small subvolumes that can be processed on the GPU. We keep two small 3D textures of size $128 \times 128 \times 128$, I_{GPU} and ϕ_{GPU}. A complete level set update is divided into a set of subvolume updates. Each subvolume in the two volumes is fetched to the GPU via 3D textures (read only, but with good cache coherence). Results of computations are written to CUDA memory and then transferred back to the CPU volumes. A simplified diagram of this is shown in Figure 2.

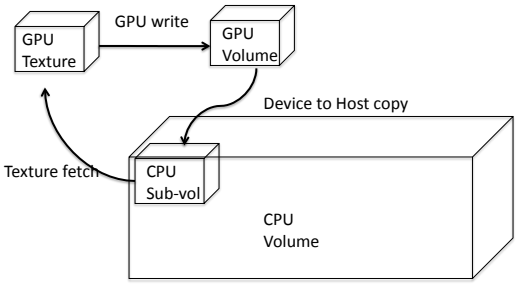


Figure 2: Streaming computation.

CUDA exposes a set of very fast 16KB shared memory available to every multi-processor in a GPU. However, a 16 KB memory chunk is shared only between a thread block, and thus to make use of it the application must load different data for different blocks. Furthermore, the 16 KB limit poses a restriction on the amount of data that can be loaded at any point of time. Here, we use 3D textures for reading the data. Since we do not want to write back to the same texture (before a single step of filtering is complete), using the read-only 3D textures available to CUDA is a natural choice. 3D texturing has hardware support for 3D cache which accelerates any texture reads in succession. To load a 3D data (a small subset of the volume) from the global memory into the shared memory could be a little tricky and might not result in the same performance as provided by the specialised hardware for 3D texture cache. In our application, data writes are made to the global memory. The latency in writing is hidden in the data processing since we do not synchronise the threads until the end of a subvolume processing. These can further be optimised by making use of coalesced writes.

The solution of the PDE is computed in iterations over the full volumes. Following are the CUDA kernels that were used in the updates.

5.1 Signed Distance Transform

Signed distance transform is a global operation and cannot be implemented in a straightforward manner. We compute a local approximation of the Euclidean distance transform using the Chamfer distance. A narrow band distance transform is computed layer by layer using, what we call a d-pass algorithm. Every pass of the method adds a layer of distance values on the existing distance transform. The distance values are local distance increments computed in a $3 \times 3 \times 3$ neighbourhood. Therefore, every single pass needs only local information to compute the distance values except at the border of the sub-volume. We therefore support every sub-volume with a one voxel cover from other adjoining sub-volumes, thereby reducing the computational domain to a volume of size $126 \times 126 \times 126$. The CPU scheduler takes care of the voxel cover. At the begining, the interface (zero level) is initialised to a used specified bounding cuboid or a super-ellipsoid.

5.2 Average Intensities

Computing average intensities (c_1 and c_2) is an operation that cannot be easily computed in a parallel fashion, and a reduction like method is required for the same. We employ a slightly different scheme to compute averages by using three accumulator sub-volumes on the GPU. These accumulators are essentially 3D sub-volumes of the same dimensions as of the textures. Every voxel in the accumulators accumulates (adds up), the values for H, $I \cdot H$, and $I \cdot (1-H)$ for all the sub-volumes in the CPU volume(s). We then sum up the small sub-volume on the CPU to get the final sum and compute c_1 and c_2 values from it. Using a mixed mode CPU-GPU computation not only reduces the complexity of an inherently non-parallel operation, but also performs better by moving less expansive parts of the computation to the CPU.

5.3 Median computation

Computing median on the GPU is not very straightforward since it is an order statistic and requires that the data be sorted. Therefore the computation of weighted

median is very different than the one for average intensity value. Since sorting values of order of millions in every iteration of the solver is not a computationally good solution, we resort to the alternative definition of the median. A median is a value that divides the data-set into two sets of equal cardinalities. This definition is generalised for a weighted median. Therefore, for a data-set V with weights W associated with each value in the set, the median value V_k is the value for which the following holds:

$$\sum_{i=0}^{k} W_i = \sum_{i=k+1}^{n} W_i$$

This equation can only be solved iteratively, starting with a guess index value k_0. In our CUDA implementation, we start with V_{k_0} to be the mean value c_1 and iteratively reach the weighted median $\hat{c}$ $(= V_k)$. In every iteration, the increment $\triangle i$ for the index k_0 is computed as:

$$\triangle i = \begin{cases} \dfrac{\sum_{i=0}^{k} W_i - \sum_{i=k+1}^{n} W_i}{\sum_{i=0}^{k} W_i}, \text{ if } \sum_{i=0}^{k} W_i > \sum_{i=k+1}^{n} W_i \\ \dfrac{\sum_{i=k+1}^{n} W_i - \sum_{i=0}^{k} W_i}{\sum_{i=k+1}^{n} W_i}, \text{ if } \sum_{i=k+1}^{n} W_i > \sum_{i=0}^{k} W_i \end{cases}$$

The increment $\triangle i$ can be adaptively controlled to give results as precise as desired.

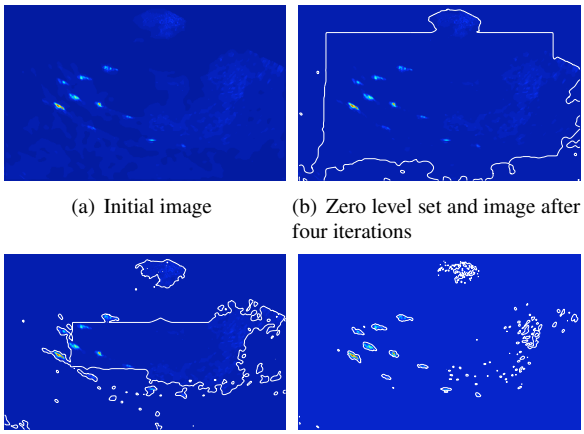
5.4 Solver update

A PDE update in the level set method comprises of computing the curvature energy and the external energy. In order to compute the curvature term (involving double derivatives) for a voxel in a sub-volume by centered differencing, we need information from a $5 \times 5 \times 5$ neighbourhood with the current voxel at its center. Therefore, the sub-volume size needs a cover of two voxels on all sides, thus reducing the computational domain further down to $124 \times 124 \times 124$. The memory schedular performs additional computations to effectively cover the whole volume with the new setup. Once the energy terms are computed, the PDE solver kernel updates ϕ_{GPU} and uses $\hat{c}$ to update I_{GPU}. These sub-volumes are then updated to the CPU main volume.

It is often convenient to perform anisotropic diffusion on the input image so that the evolution of the level curve is smooth and ϕ is well behaved. Finally, the zero level surface is extracted from the evolved ϕ using the Marching-cubes method.

6 EXPERIMENTAL RESULTS

We present experimental results on sample acoustic 2D images to show that the suppression scheme works well on such images. Figure 3 shows evolution of the level set. The parameters for this evolution were chosen to be: $\mu = 0.0005$, $\nu = 0$, $\lambda_1 = \lambda_2 = 1$, and $\varepsilon = 2.5$. It can be seen that the original image suffers from speckle noise as seen in figure 4 and that the final zero level contour approximates the fish boundaries very well.



(a) Initial image

(b) Zero level set and image after four iterations

(c) Zero level set and image after ten iterations

(d) End of evolution after 16 iterations

Figure 3: Level set evolution on sample image. $\varepsilon = 2.5$.

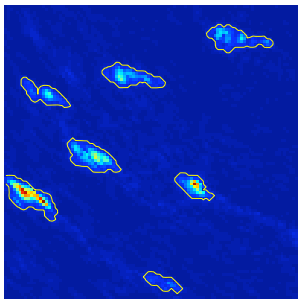
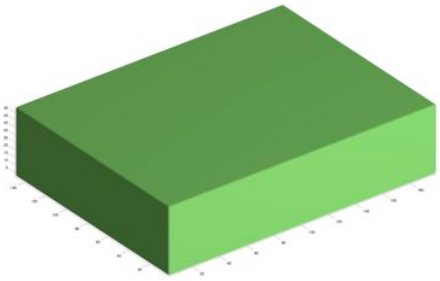


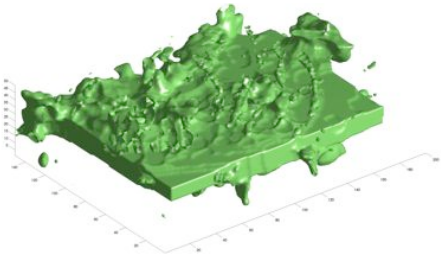
Figure 4: The final contour shown on the part of the original image.

We next show results of application of the level set equation and ths noise suppression scheme on a small 3D volume of size $150 \times 100 \times 50$. Fish intensities can be identified in dark green against a noisy background. The level set equation was initialised with the zero level set of ϕ_0 as the bounding box of the volume. The level set is then allowed to evolve with parameters, $\mu = 0.0005$, $\nu = 0$, $\lambda_1 = \lambda_2 = 1$, and $\varepsilon = 1.0$. Figure 5 shows the evolution at different time steps and the final level surface.

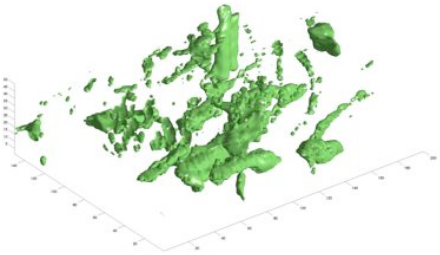
We test the CUDA solver on a larger volume of size $686 \times 1234 \times 100$. This volume uses about 470 MB of CPU memory along with the same amount of memory consumed by the signed distance field. Figure 6 shows the extracted fish trails. We test our implementation



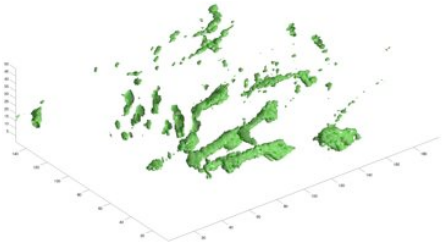
(a) Initial zero level surface, ϕ_0



(b) Zero level surface after four iterations



(c) Zero level surface after six iterations



(d) End of evolution after nine iterations

Figure 5: Level set evolution on sample volume. $\varepsilon = 1.0$.

with the mobile GPU, GeForce 8600M GT (NVIDIA CUDA compute capability of 1.1) with 256 MB of memory on a Mac OS X notebook. The total number of iterations required until convergence were 29, with a compute time of about 67 seconds per iteration. Similar test on a faster GPU, GeForce 8800 GTX with 768 MB of memory yielded a compute time of 25 seconds per iteration (see Table 1 for computation times for various solver operations). The signed distance field was reconstructed in a narrow band of width 20 voxels in every iteration. With the commodity graphics hardware, we expect to get better speedups. Furthermore, a better GPU with more onboard memory should allow loading larger subvolumes, thus reducing the overhead of multiple memory transfers.

GPU→	8600M GT	8800 GTX
Signed Distance	39.52 sec	9.47 sec
Average Intensity	3.17 sec	1.10 sec
Weighted Median	18.92 sec	11.45 sec
PDE Update	5.63 sec	3.37 sec

Table 1: Computation times for various operations tested on two GPUs. Processed volume size is $686 \times 1234 \times 100$.

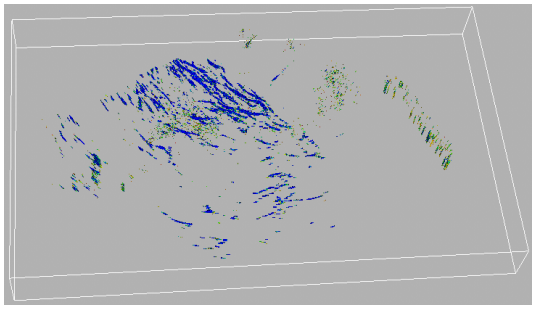


Figure 6: Fishes extracted from a volume of size $686 \times 1234 \times 100$.

In order to compare the 2D and 3D reconstructions, we show an overlay of 2D curves over the extracted 3D surface. This is shown in figure 7. The results agree very well when the 2D image contains high intensity objects. The acoustic images were taken by scanning fishes in an aquarium and the images corresponding to the bottom of the aquarium (time slices with higher depth, 30 to 50 in figure 7) contain almost no fishes. Therefore, these images contain very little useful information. The 2D level set evolution fails to detect fishes in these images. It is also worth mentioning that the suppression of noise is based on weighted median and if the images do not contain high intensities, it is possible that the estimated threshold value does not accurately represent the noise level. Therefore, the 3D results should be trusted since the 2D reconstruction does not consider information present in other image planes. We would like to comment that a ground truth segmentation is not practically possible for open sea. Evaluation of the extracted fish trails/schools by domain experts is under process because of marine surveys.

While we claim that this method works on acoustic images with high variance in intensity values resulting in a binary segmentation of the image, it is certainly possible to perform a segmentation resulting in more than two segments [2].

7 CONCLUSIONS

In this paper, we presented augmentation of level set formulation based on the Mumford-Shah functional to a

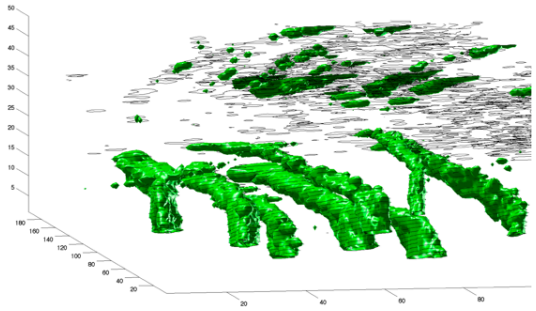


Figure 7: Comparison of zero level 2D curves with the zero level 3D surface.

noise suppression scheme, well suited for object reconstruction from acoustic images. Our method is based on computation of a threshold by weighted median of intensity values. We prove that the method converges with evolving level set and show that the experimental results comply with that. We show a 3D reconstruction of objects from time series images which is useful in tracking moving objects and to observe their kinetics. An optimised GPU based implementation has been presented for streaming computation of the large volumetric data.

ACKNOWLEDGEMENTS

The acoustic data was collected in collaboration with the Danish Institute of Fisheries Research (DIFRES) in Denmark, and the authors are thankful for their support.

REFERENCES

[1] J. P. Balabanian, I. Viola, E. Ona, R. Patel, and M. E. Gröller. Sonar explorer: A new tool for visualization of fish schools from 3d sonar data. In *Data Visualization - EuroVis 2007*, pages 155–162. IEEE, 5 2007.

[2] T.F. Chan and L. Vese. A level set algorithm for minimizing the Mumford-Shah functional in image processing. *IEEE/Computer Society Proceedings of the 1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 161–168, 2001.

[3] T.F. Chan and L.A. Vese. Active Contours Without Edges. *IEEE Transactions on Image Processing*, 10(2), 2001.

[4] Y. Chen and A. Raheja. Wavelet Lifting for Speckle Noise Reduction in Ultrasound Images. *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 3129–3132, 2005.

[5] J. Dunlop. Statistical modelling of sidescan sonar images. *OCEANS'97. MTS/IEEE Conference Proceedings*, 1, 1997.

[6] V. Dutt and J. F. Greenleaf. Adaptive speckle reduction filter for log-compressed B-scan images. *Medical Imaging, IEEE Transactions on*, 15(6):802–813, 1996.

[7] L. Gagnon. Wavelet filtering of speckle noise-some numerical results. *Proc. on the Conference Vision Interface ('99), Trois-Rivieres, Canada*, pages 1–8, 1999.

[8] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, 2007.

[9] N. K. Govindaraju, B. Lloyd, W. Wang, M. Lin, and D. Manocha. Fast computation of database operations using graphics processors. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 206, New York, NY, USA, 2005. ACM.

[10] A. Isar, D. Isar, S. Moga, J.M. Augustin, and X. Lurton. Multiscale MAP despeckling of sonar images. *Oceans 2005-Europe*, 2, 2005.

[11] K. Krissian, K. Vosburgh, R. Kikinis, and C.F. Westin. Anisotropic diffusion of ultrasound constrained by speckle noise model. *Department of Radiology, Brigham and Women's Hospital, Harvard Medical School, Laboratory of Mathematics in Imaging, Tech. Rep*, 4, 2004.

[12] A. E. Lefohn, J. M. Kniss, C. D. Hansen, and R. T. Whitaker. A streaming narrow-band algorithm: Interactive computation and visualization of level sets. *IEEE Transactions on Visualization and Computer Graphics*, 10:422–433, 2004.

[13] J. Li, C. A. Papachristou, and R. Shekhar. A "brick" caching scheme for 3d medical imaging. In *ISBI*, pages 563–566, 2004.

[14] M. Lianantonakis and Y. R. Petillot. Sidescan sonar segmentation using active contours and level set methods. *Oceans 2005-Europe*, 1, 2005.

[15] T. Loupas, WN McDicken, and P. L. Allan. An adaptive weighted median filter for speckle suppression inmedical ultrasonic images. *Circuits and Systems, IEEE Transactions on*, 36(1):129–135, 1989.

[16] R. Malladi and J. A. Sethian. Image Processing Via Level Set Curvature Flow. *Proceedings of the National Academy of Sciences of the United States of America*, 92(15):7046–7050, 1995.

[17] M. Mignotte, C. Collet, P. Perez, and P. Bouthemy. Unsupervised Markovian segmentation of sonar images. *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, 4, 1997.

[18] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.*, 42(5):577–685, 1989.

[19] S. Osher and R.P. Fedkiw. *Level sets and dynamic implicit surfaces*. Springer New York, 2003.

[20] S. Osher and L.I. Rudin. Feature-Oriented Image Enhancement Using Shock Filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, 1990.

[21] S. J. Osher and J.A. Sethian. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[22] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[23] I. Quidu, J. P. Malkasse, G. Burel, and P. Vilbé. A 2-D Filter Specification for Sonar Image Thresholding; 2001. *Advanced Concepts for Intelligent Vision Systems (ACIVS'2001) conference, Baden-Baden, Germany*, 2001.

[24] J.A. Sethian. Theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numerica 1996*, pages 309–395, 1996.

[25] E.J. Simmonds and D.N. MacLennan. *Fisheries Acoustics: Theory and Practice*. Blackwell Publishing, 2005.

[26] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo. Weighted median filters: a tutorial. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43(3):157–192, 1996.